

(19)

Europäisches Patentamt

European Patent Office

Office européen des brevets

B3



(11)

EP 0 793 171 A1

(12)

DEMANDE DE BREVET EUROPEEN

(43) Date de publication:
03.09.1997 Bulletin 1997/36

(51) Int Cl.⁶: G06F 9/445

(21) Numéro de dépôt: 97400441.8

(22) Date de dépôt: 27.02.1997

(84) Etats contractants désignés:
BE DE ES FR GB IT LU NL

(30) Priorité: 01.03.1996 FR 9602641

(71) Demandeur: BULL S.A.
78430 Louveciennes (FR)

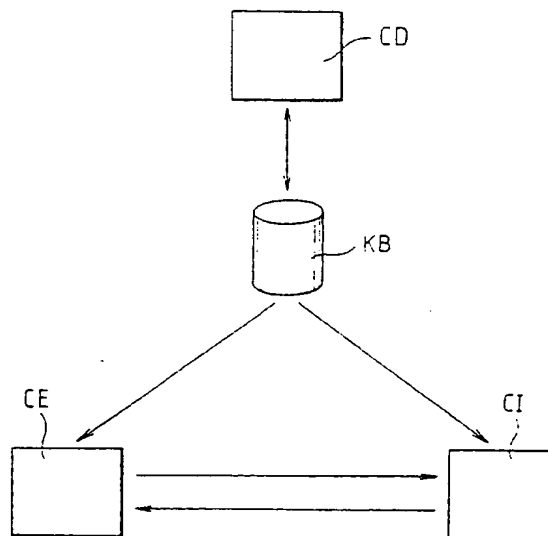
(72) Inventeurs:
• Gibello, Pierre-Yves
38100 Grenoble (FR)
• Albrecht, Alain
38400 Saint Martin D'Heres (FR)

• Richoux, Bruno
38410 Uriage (FR)
• Fleurisson, Marc
38250 Lans En Vercors (FR)
• Sehabiague, Bruno
38250 Saint Nizier Du Moucherotte (FR)

(74) Mandataire: Debay, Yves
Cabinet Yves Debay,
122 Elysée 2
78170 La Celle Saint Cloud (FR)

(54) **Système de configuration de logiciels préconfigurés sur des systèmes ouverts en réseau dans un environnement distribué et procédé mis en oeuvre par un tel système**

(57) La présente invention concerne un système de configuration de logiciels sur des systèmes ouverts en réseau dans un environnement distribué. Ce système est remarquable en ce qu'il est constitué, d'une part sur une station de travail, d'un dispositif de conception appelé concepteur (CD), d'un éditeur (CE) et d'une base de connaissances (KB), et d'autre part sur une station à configurer d'un interpréteur (CI), le concepteur permettant d'accéder en lecture et écriture à la base de connaissances contenant les données décrivant les différents éléments du réseau qui peuvent être configurés et les liaisons entre ces différents éléments, ledit concepteur permettant de définir les différentes possibilités de configuration de l'éditeur sur lequel est visualisée la configuration désirée. L'éditeur utilise les informations de la base de connaissances relatives à la topologie du réseau et les paramètres de configuration des différents éléments composant le réseau pour construire une entité homogène configurable et chargeable, à l'aide de moyens de chargement, sur la station à configurer dans laquelle l'interpréteur exécute les opérations de configuration en transformant les paramètres de configuration initiaux en des paramètres propres à la configuration désirée également utilisables par l'éditeur. Pour cela, ce système met en oeuvre un procédé spécifique de configuration de logiciels préconfigurés.



EP 0 793 171 A1

Description

Konfiguration auf logisch Ebene
La présente invention concerne un système de configuration de logiciels sur des systèmes ouverts en réseau dans un environnement distribué. Elle concerne également un procédé mis en oeuvre par ledit système.

De manière générale, en informatique et en particulier avec les réseaux complexes de systèmes distribués, il est apparu indispensable de prévoir et organiser des ensembles de services d'utilisation simple, conviviale et efficace et notamment proposer des services de configuration de logiciels sur les plates-formes de cet environnement réparti qui soient puissants mais souples et rapides à mettre en oeuvre. Un tel environnement est en effet, de nos jours, couramment exploité et, il est connu qu'utiliser une informatique ouverte et distribuée présente d'énormes avantages puisqu'il est alors permis à un client, c'est-à-dire l'utilisateur d'une interface, d'accéder aisément aux différentes données de tels systèmes ou machines quelle que soit la situation géographique dudit utilisateur et quelle que soit le système sur lequel résident les données. Cependant, si l'utilisateur désire profiter de ces avantages durant la phase d'exploitation, il souhaite également, que les phases de configuration et d'installation soient aussi facilitées et ceci pour toutes sortes de déploiements. Par configuration, il est ici entendu la manipulation de paramètres d'une machine tels que l'adresse, le nom de l'utilisateur, la taille des registres tampons, les données relatives aux paramètres de configuration des logiciels, etc.

Jusqu'à présent, des outils tels que SMIT (System Management Interface Tool) ou DSMIT (Distributed System Management Interface Tool) étaient utilisés pour configurer des logiciels sur des plates-formes UNIX (marque déposée aux US et dans d'autres pays, licenciée exclusivement par l'intermédiaire de X/OPEN Company Limited) supportant le système d'exploitation AIX (marque d'IBM Corp.). Cependant, à l'usage, il apparaît que ces outils présentent d'importants inconvénients. En effet, si l'outil SMIT permet de travailler pendant les phases d'installation, de configuration et d'exploitation, son utilisation est néanmoins limitée aux systèmes locaux, tels qu'un serveur et donc à une seule machine, n'étant pas conçu pour prendre en compte, ni l'aspect réseau, ni l'aspect distribué, ce qui le rend totalement inefficace pour des applications relatives à des systèmes en réseau en environnement distribué. L'outil DSMIT, quant à lui, a les mêmes caractéristiques que l'outil SMIT à ceci près cependant, qu'il permet de prendre en compte l'aspect distribué, mais il s'avère inefficace pour traiter l'aspect réseau. Ainsi avec l'outil DSMIT il sera, par exemple, possible de configurer plusieurs machines en même temps mais avec une même adresse, ce qui est, il faut en convenir, très rarement le cas dans un réseau.

Pour configurer des logiciels de communication sur des plates-formes, une autre solution peut également

être utilisée qui consiste à appliquer une méthode de répartition ou de distribution de logiciel couramment appelée par l'homme du métier "software distribution". Une telle méthode est généralement utilisée par des administrateurs de systèmes de gestion ou d'administration centralisés pour diffuser des logiciels sur différentes plates-formes administrées. Dans ce cas, il est possible de réaliser une installation et à la fin de cette installation d'émettre une information (appelée "script" par l'homme du métier et qui est en fait un petit programme de personnalisation) renfermant des instructions de configuration. Cependant, pour autoriser une telle configuration, cette information qui n'est pas créée ou construite automatiquement, doit préalablement être décrite et entrée par un utilisateur, ce qui manque assurément de simplicité et de convivialité. De plus, la syntaxe de ces scripts doit obéir à des règles strictes et contraignantes, notamment pour soutenir la caractéristique de ré-entrance. La ré-entrance fait, en effet, appel à une notion intelligente, car le script, pour cela, doit savoir distinguer entre un premier lancement et un lancement subséquent, sachant que le résultat doit rester invariant, tandis que l'action dépend de l'état initial de l'environnement au lancement du script.

La présente invention a pour but de remédier aux divers inconvénients des différents systèmes ou outils et méthodes de l'art antérieur connus et propose un système de configuration de logiciels sur des systèmes ouverts, convivial, et offrant une solution, globale, efficace, automatique et donc aisée à mettre en oeuvre, aux problèmes posés par la configuration de systèmes en réseau dans un environnement distribué.

Pour cela, le système de configuration de logiciels mentionné dans le préambule est remarquable en ce qu'il est constitué, d'une part sur une station de travail, d'un dispositif de conception appelé concepteur, d'un éditeur et d'une base de connaissances, et d'autre part sur une station à configurer d'un interpréteur, le concepteur permettant d'accéder en lecture et écriture à la base de connaissances contenant les données décrivant les différents éléments du réseau qui peuvent être configurés et les liaisons entre ces différents éléments, ledit concepteur permettant de définir les différentes possibilités de configuration de l'éditeur sur lequel est visualisée la configuration désirée, éditeur qui utilise les informations de la base de connaissances relatives à la topologie du réseau et les paramètres de configuration des différents éléments composant le réseau pour construire une entité homogène configurable et chargeable, à l'aide de moyens de chargement, sur la station à configurer dans laquelle l'interpréteur exécute les opérations de configuration en transformant les paramètres de configuration initiaux en des paramètres propres à la configuration désirée également utilisables par l'éditeur, ledit système mettant en oeuvre un procédé de configuration de logiciels préconfigurés.

De manière préférée le système de configuration de logiciels préconfigurés est remarquable en ce que le

concepteur utilise une interface utilisateur graphique du type "Windows" qui permet de définir de nouvelles classes de plates-formes et de nouvelles classes de composants de systèmes pour étendre les informations de la base de connaissances utilisées par l'éditeur, en offrant pour cela la possibilité de définir, modifier ou supprimer des classes de plates-formes, également de définir, modifier ou supprimer des classes de composants de systèmes et les feuilles d'informations associées à ces classes et enfin relativement aux classes de composants, de définir, modifier ou supprimer des informations des classes de plates-formes relatives à la configuration désirée.

De même, le système de configuration de logiciels préconfigurés est remarquable en ce que l'éditeur utilise une interface utilisateur graphique du type "Windows" qui, en premier lieu, fournit des moyens graphiques à l'utilisateur pour construire à l'écran une topologie du réseau représentant l'ensemble des systèmes hétérogènes avec leurs interconnexions, pour chaque système représenté une feuille ou un ensemble de feuilles d'informations de configuration étant rempli, qui également fournit des mécanismes de sauvegarde, d'ouverture et de restauration des configurations définies par l'utilisateur ainsi que des mécanismes de gestion de l'unicité d'un objet à travers le réseau représenté, qui de plus, pour une configuration spécifiée par l'utilisateur, génère un ensemble de fichiers de configuration dans un format déterminé, ces fichiers étant utilisés pour générer une information de configuration qui est exécutée sur la plate-forme désirée configurée et qui enfin, à partir d'un ensemble de fichiers de configuration récupéré sur une plate-forme configurée, extrait la configuration réelle désirée de cette plate-forme.

Egalement, le système de configuration de logiciels préconfigurés est remarquable en ce que l'interpréteur d'une part, lit les fichiers de configuration générés par l'éditeur dans le format déterminé en faisant la différence entre une première configuration, une reconfiguration ou une déconfiguration, puis convertit lesdits fichiers en directives de configuration pour la plate-forme sur laquelle il est situé et enfin effectue la configuration désirée de cette plate-forme et d'autre part, lit la configuration effective de la plate-forme, puis génère les fichiers de configuration dans le format déterminé contenant les informations de configuration des composants de cette plate-forme, lesdits fichiers de configuration étant alors utilisables par l'éditeur.

Ainsi ce système qui s'inscrit dans une technologie orientée-objet autorise une configuration automatique de logiciels sur des serveurs ou machines travaillant dans des environnements complexes en mode client-serveur selon une architecture distribuée et offre un service de configuration global aux utilisateurs désirant déployer rapidement un grand nombre de systèmes à partir d'un seul point. En effet, selon l'idée de l'invention, à partir d'un seul point, d'une même console d'une station de travail, il est possible de préparer une configuration

de référence qui peut être dupliquée sur différentes machines, ou de configurer de manière spécifique une quelconque station, c'est-à-dire machine ou plate-forme, d'un ensemble hétérogène sur le réseau, l'interpréteur, de manière intelligente, à la différence de ce qui se passait dans l'art antérieur, se chargeant de discriminer automatiquement entre les paramètres initiaux de configuration et les paramètres propres à la configuration désirée en choisissant de manière adéquate ces derniers. Cette configuration désirée de la plate-forme est ici préparée sur la console d'un microordinateur en utilisant une application conviviale du type "Windows" pour remplir une ou plusieurs fiches ou "feuilles" d'informations sur l'écran relative à la topologie et aux différents paramètres désirés, et ceci tout simplement par sélection sur l'écran pour dessiner la topologie du réseau, fixer des paramètres tels que le site, la machine, les connexions, les adresses, les ports, les options, etc., et ainsi créer l'image de la configuration souhaitée, diffusée, après génération d'un fichier de configuration, vers la machine sur laquelle elle sera chargée pour être exécutée, l'interpréteur permettant ce chargement et faisant avantageusement la différence entre ce qui existe et ce qui est désiré, alors qu'en outre, cette configuration peut être contrôlée et consultée à partir du microordinateur. Il peut être ici insisté sur "l'intelligence" de l'interpréteur qui est capable d'identifier la configuration présente et en fonction de la configuration transmise désirée, de reconnaître s'il s'agit d'une première configuration, d'une reconfiguration ou encore d'une déconfiguration pour effectuer alors la commande appropriée. Il est aussi possible de configurer de manière automatique des serveurs similaires pour compléter ou modifier une infrastructure, simplement à partir de moyens de chargement tels qu'une disquette, des moyens électroniques utilisant le transfert de fichiers, ou encore des services communs sur un réseau du type TCP/IP (Transmission Control Protocol/Internet Protocol). Grâce à ce nouveau concept homogène et centralisé de configuration par commande, l'ouverture à de nouveaux objets configurables est autorisée facilitant tout déploiement ou migration aujourd'hui non permis, y compris le déploiement d'applications d'un client intégrateur de service. Cette dernière caractéristique d'extensibilité est rendue possible du fait que des dialogues de configuration et des commandes spécifiques peuvent être aisément créés et qu'ainsi de nouveaux objets peuvent être ajoutés via le concepteur dont c'est précisément l'objet. Une telle avantageuse convivialité était jusqu'à présent interdite aux réseaux de systèmes hétérogènes en environnement distribué. De nombreux avantages se dégagent du fait de la matérialisation de l'idée de l'invention, parmi lesquels l'extrême simplicité d'utilisation et d'application des procédures de configuration, tandis qu'une meilleure visibilité de la configuration installée est offerte, le graphique de ladite configuration étant directement affichée sur l'écran. Egalement, le temps d'apprentissage pour l'utilisation de telles procédures

ainsi donc que le savoir-faire nécessaire à leur exécution peuvent être considérablement réduits et ceci même relativement à un large déploiement de systèmes à, ou à ne pas, dupliquer dans des sites aléatoirement et géographiquement distribués. De plus, la productivité et l'efficacité sont très nettement améliorées puisque l'administrateur permet de configurer facilement et rapidement les systèmes en réseau sans oublier que l'automatisation du processus et les possibilités de duplication sont bien supérieures permettant ainsi de réduire les coûts et les délais. Une configuration globale cohérente associée au fait qu'une plus faible technicité est requise pour installer le logiciel implique nécessairement, pour l'administrateur, un meilleur contrôle de la gestion du déploiement des systèmes, ce qui a pour principal effet d'améliorer la fiabilité de l'ensemble. En outre, une très grande autonomie est autorisée à un client utilisateur qui, achetant un progiciel, pourra l'installer lui-même et ceci de manière immédiate, car il n'aura plus besoin systématiquement d'une assistance technique. Enfin, avantageusement, dans cet environnement complexe en mode client-serveur, l'éditeur fournit des mécanismes puissants et pratiques tels que notamment des mécanismes de gestion de l'unicité d'un objet et donc de données à travers le réseau représenté, ainsi, par exemple, une adresse d'un serveur distribuée sur le réseau peut être mise à jour simultanément sur l'ensemble dudit réseau.

Pour cela, de manière avantageuse, le système selon l'invention met en oeuvre un procédé de configuration de logiciels préconfigurés remarquable en ce qu'il est divisé en une pluralité de sous-procédés, un sous-procédé d'édition de la base de connaissances permettant à un utilisateur d'une station de travail d'accéder à la base de connaissances pour créer, éditer, modifier et enregistrer ladite base, un sous-procédé de détermination de la configuration permettant à un utilisateur d'une station de travail de construire une configuration désirée, un sous-procédé d'actualisation de la configuration désirée sur la station à configurer, un sous-procédé d'analyse et de contrôle de la configuration sur la station configurée et enfin un sous-procédé d'exploration de la configuration d'une station configurée à partir d'une station de travail.

Ainsi le procédé de configuration mis en oeuvre par le système selon l'invention est fractionné en plusieurs sous-procédés. Un premier sous-procédé d'édition de la base de connaissances qui est indépendant des autres sous-procédés et qui permet de construire une base de connaissances, de la modifier et de la faire évoluer à volonté. Un second sous-procédé de détermination de la configuration qui, une fois exécuté sur la station de travail est suivi du troisième sous-procédé d'actualisation de la configuration pour réaliser effectivement par et sur la plate-forme à configurer ladite configuration. Enfin, un quatrième sous-procédé d'analyse et de contrôle de la configuration sur une station configurée, qui une fois exécuté sur cette station est suivi

d'un cinquième sous-procédé d'exploration de la configuration de la station configurée à partir d'une station de travail permettant d'extraire et de récupérer effectivement la configuration de cette station configurée.

La description suivante en regard du dessin annexé, le tout donné à titre d'exemple non limitatif, fera bien comprendre comment l'invention peut être réalisée.

La figure unique représente de manière schématique les éléments formant le système de configuration de logiciels préconfigurés sur des systèmes ouverts selon l'invention.

Pour une meilleure appréhension de l'idée de l'invention, quelques généralités et définitions sont ci-après proposées qui permettront de replacer ladite invention dans son contexte et de comprendre avec précision le vocabulaire employé. Le système de configuration de logiciels préconfigurés a été conçu pour apporter une solution simple, peu onéreuse et efficace aux problèmes complexes rencontrés lorsqu'il est désiré réaliser la migration et/ou le déploiement ainsi que la mise à jour d'un grand nombre, de préférence supérieur ou égal à dix, de systèmes hétérogènes en réseau dans un environnement distribué. Ce système de configuration permet donc de déployer rapidement et de manière aisée ce grand nombre de systèmes ou serveurs interconnectés car il fournit un environnement de configuration autorisant la préparation de la configuration de logiciels hors du ou des sites administrés sur lesquels se trouvent les systèmes à configurer en offrant des possibilités de duplications sur la pluralité de sites concernés et de paramétrisations répétitives et automatiques. Dans la suite, le terme "composant" doit être assimilé à une entité physique ou logique d'un système informatique telle que du matériel, un logiciel ou encore un microprogramme, l'objet du système selon l'invention étant de configurer des composants. Un composant est défini par une "classe" constituée d'un nom unique et d'un ensemble d'attributs qui, en commun, définissent l'information de configuration disponible, chaque attribut représentant une partie de l'information de configuration. Les attributs décrivant la configuration d'une même fonctionnalité sont réunis en groupes dits fonctionnels pour être facilement référencés. La feuille ou fiche d'informations de configuration est, pour l'utilisateur, la vue, c'est-à-dire la matérialisation sur l'écran, d'un groupe fonctionnel, et, c'est en fait, à l'aide de cette feuille et en la remplissant, que l'utilisateur précise les informations nécessaires à la configuration d'un groupe fonctionnel d'un composant. Les composants définissent la configuration des systèmes hétérogènes tels que les unités centrales de traitement, les stations de travail de type PC ou les plates-formes UNIX. Chaque système peut comporter un nombre quelconque de composants de différentes classes. Chaque groupe fonctionnel d'une classe de composants peut être instancié plus d'une fois, pour décrire des instances multiples d'une partie d'un composant. Par convention, un

groupe fonctionnel est dit, "scalaire" lorsqu'il n'est instancié qu'une seule fois, ou "tabulaire" lorsqu'il peut être instancié plus d'une fois. Les différents systèmes sont connectés à leur réseau d'appartenance par l'intermédiaire de "liaisons" soit point à point (par exemple, liaison X25) soit multipoint ou fédératives (par exemple, Ethernet, FDDI, "token ring"). Pour préciser, une liaison point à point permet de connecter un système à un autre système ou un système à une liaison fédérative, alors qu'une liaison fédérative est connectée à un ensemble de systèmes par l'intermédiaire d'une liaison point à point.

Sur la figure unique sont représentés les différents éléments composant le système de configuration de logiciels préconfigurés conforme à l'invention. Ce système est constitué, d'une part sur une station de travail (non représentée), d'un dispositif de conception CD appelé concepteur, d'un éditeur CE et d'une base de connaissances KB, et d'autre part sur une station à configurer (non représentée) d'un interpréteur CI, le concepteur CD permettant d'accéder en lecture et écriture à la base de connaissances KB contenant les données décrivant les différents éléments du réseau qui peuvent être configurés et les liaisons entre ces différents éléments (systèmes, composants, groupes fonctionnels, etc.), ledit concepteur permettant de définir les différentes possibilités de configuration de l'éditeur CE sur lequel est visualisée la configuration désirée, éditeur qui utilise les informations de la base de connaissances KB relatives à la topologie du réseau et les paramètres de configuration des différents éléments composant le réseau pour construire une entité homogène configurable et chargeable, à l'aide de moyens de chargement (disquette, moyens électroniques utilisant le transfert de fichiers, ou encore services communs sur un réseau du type TCP/IP), sur la station à configurer dans laquelle l'interpréteur exécute les opérations de configuration en transformant les paramètres de configuration initiaux en des paramètres propres à la configuration désirée également utilisables par l'éditeur CE, ledit système mettant en oeuvre un procédé de configuration de logiciels préconfigurés.

Dans cette perspective, le concepteur CD se trouvant sur la station de travail de type PC, est conçu comme une interface graphique utilisateur de type Windows et permet de déterminer de manière simple ce qui peut être configuré par l'éditeur CE. En fait, l'éditeur CE exploite un ensemble de ressources bien défini, alors que l'utilisateur, pendant que l'éditeur exécute ses tâches de configuration, manipule d'une part les classes de plates-formes, chaque classe de plates-formes ayant pour objet de définir un type de système, et d'autre part les composants. L'utilisateur manipule les composants par l'intermédiaire des feuilles d'informations de configuration, les définitions des composants étant stockées dans les classes de composants. Une fonction importante concerne les possibilités d'extension du système de configuration, par ajout, modification ou suppression d'objets

dans la base de connaissances KB, le concepteur permettant de définir de nouvelles classes de plates-formes et de nouvelles classes de composants autorisant ainsi l'extension de la base de connaissances KB utilisée par l'éditeur CE. Pour cela, le concepteur CD offre différentes possibilités parmi lesquelles celles de définir, modifier ou supprimer des classes de plates-formes, et également de définir, modifier ou supprimer des classes de composants de systèmes et les feuilles d'informations associées à ces classes, ces informations étant principalement utilisées par l'éditeur CE. Enfin, relativement aux classes de composants, il offre la possibilité de définir, modifier ou supprimer des informations concernant les mécanismes de configuration désirée agissant sur les classes de plates-formes, ces informations étant principalement utilisées par les interpréteurs CI sur les stations à configurer. Ce type de mécanisme, appelé ci-après "instrumentation de composant", spécifie la manière de configurer une classe de composants et correspond en fait à une interface utilisée pour fixer (set), test d'installation ou de robustesse, ou lire (get), test d'inventaire ou d'analyse des informations de configuration relatives à un composant matériel ou logiciel ou encore à une feuille d'information concernant un groupe fonctionnel scalaire ou tabulaire. Il est également important de noter que la notion de composant logiciel configurable est à rapprocher de celle de produit logiciel installable.

De même, l'éditeur CE se trouvant sur la station de travail de type PC, est conçu comme une interface graphique utilisateur de type Windows et permet de traiter les informations de configuration relativement à un ensemble de composants défini. Ces principales fonctionnalités consistent en premier lieu, à fournir des moyens graphiques à l'utilisateur pour construire à l'écran une topologie du réseau contenant un quelconque ensemble de systèmes hétérogènes interconnectés, et pour chaque système représenté, une feuille ou un ensemble de feuilles d'informations de configuration est rempli avec les informations relatives aux composants faisant partie de ce système. Le résultat de cette opération de construction visualisable est la configuration (systèmes et liaisons). L'éditeur est également prévu pour fournir des mécanismes de sauvegarde, d'ouverture et de restauration des configurations définies par l'utilisateur ainsi que des mécanismes de gestion de l'unicité d'un objet à travers le réseau représenté autorisant à ne rentrer qu'une et une seule fois une même partie d'information de configuration. De plus, pour une configuration spécifiée par l'utilisateur, l'éditeur permet de générer un ensemble de fichiers de configuration dans un format déterminé. De manière préférée, le fichier utilisé est de format MIF (Management Interface Format) normalisé par le consortium DMTF (Desktop Management Task Force), format qui permet de décrire des composants matériel ou logiciel en termes d'objets et d'attributs. Ces fichiers sont utilisés pour générer une information de configuration qui est exécutée sur la plate-forme désirée

configurée. Il est à noter que bien que ces informations de configuration sont différentes d'une plate-forme à une autre, les mêmes fichiers générés par l'éditeur, pour une configuration donnée, sont utilisés pour générer les informations de configuration pour toutes les plates-formes concernées par cette configuration. Enfin, à partir d'un ensemble de fichiers de configuration récupéré sur une plate-forme configurée, l'éditeur permet d'extraire la configuration réelle de cette plate-forme, c'est-à-dire, que les feuilles d'informations de configuration des composants de la plate-forme peuvent être éditées en utilisant l'éditeur et que, lorsqu'elle est introduite dans une configuration, la plate-forme est connectée de manière adéquate aux autres éléments de cette configuration, ceci étant déduit des données de configuration de cette plate-forme. L'éditeur fait la distinction entre une première édition d'un nouvel objet (plate-forme, produit, feuille d'informations) et donc la création d'une représentation graphique ou la mise à jour de cette représentation graphique en prenant en compte les nouveaux paramètres ou encore la destruction de l'un ou l'autre, composant ou feuille d'informations, si ces derniers sont absents dans les nouveaux paramètres.

Egalement, l'interpréteur CI se trouvant sur la station à configurer utilise les informations et les fonctions offertes par la base de connaissances BK et, indépendamment du type de plate-forme, présente une fonctionnalité de lecture double. D'une part, il lit les fichiers de configuration générés par l'éditeur dans le format déterminé, de préférence dans le format MIF, en faisant la différence entre une première configuration, une reconfiguration ou une déconfiguration, puis convertit lesdits fichiers en directives de configuration pour la plate-forme sur laquelle il est situé et ensuite effectue la configuration désirée de cette plate-forme. D'autre part, il lit la configuration effective de la plate-forme, puis génère les fichiers de configuration dans le format déterminé, de préférence dans le format MIF, contenant les informations de configuration des composants de cette plate-forme, lesdits fichiers de configuration étant alors utilisables par l'éditeur. Lors d'une configuration, l'interpréteur peut, si cela est nécessaire et désiré, renommer la machine pour la personnaliser, c'est-à-dire lui donner un nom différent de celui donné lors de sa sortie de fabrication et également, si cela est désiré, relancer la machine (commande UNIX "reboot") en fin de configuration.

Dans un mode de réalisation préféré, la station de travail sur laquelle sont implantés le concepteur CD et l'éditeur CE est une plate-forme du genre PC ou compatible PC supportant au moins un environnement système de type Windows 3.1, alors que la station à configurer comportant l'interpréteur CI est une plate-forme UNIX.

Avantageusement, le système selon l'invention met en oeuvre un procédé de configuration de logiciels pré-configurés remarquable en ce qu'il est divisé en une pluralité de sous-procédés, un sous-procédé d'édition de

la base de connaissances permettant à un utilisateur d'une station de travail d'accéder à la base de connaissances pour créer, éditer, modifier et enregistrer ladite base, un sous-procédé de détermination de la configuration permettant à un utilisateur d'une station de travail de construire une configuration désirée, un sous-procédé d'actualisation de la configuration désirée sur la station à configurer, un sous-procédé d'analyse et de contrôle de la configuration sur la station configurée et enfin un sous-procédé d'exploration de la configuration d'une station configurée à partir d'une station de travail.

Une définition plus précise de la base de connaissances et en particulier de son contenu fonctionnel permettra de mieux comprendre comment les données internes sont manipulées par les sous-procédés mis en oeuvre par le système de configuration, lesdites données contenues dans la base de connaissances décrivant les entités qui peuvent être configurées ainsi que les relations entre ces entités. Suivent ainsi les explications relatives à une pluralité d'entités et à leurs interconnexions que la base de connaissances permet de décrire.

Un quelconque système informatique peut être modélisé au moyen d'une classe de plates-formes conçue pour identifier des plates-formes qui ont des caractéristiques de configuration similaires, sachant qu'une classe de plates-formes peut être définie par:

- un nom qui l'identifie de manière unique au sein de la base de connaissances, ce nom identifiant par exemple un type de machines et cette information de nom étant obligatoire,
- un commentaire extérieur lié à la classe de plates-formes, l'utilisateur de l'éditeur pouvant accéder à ce commentaire, cette information étant optionnelle,
- un commentaire interne lié à la classe de plates-formes, seul l'utilisateur du concepteur pouvant accéder à ce commentaire qui est interdit à l'utilisateur de l'éditeur, cette information étant optionnelle,
- un icône graphique représentant la classe de plates-formes, cette information étant optionnelle.

Le produit ou l'élément à configurer est vu comme un composant défini par la classe à laquelle il appartient. Pour cela, une classe de composants est entièrement définie par:

- un nom, information qui est obligatoire,
- un numéro de version, de forme VRMF (Version Release Maintenance Fix) qui permet d'identifier la version du composant selon un format prédéterminé et normalisé, cette information étant obligatoire. Une classe de composants est identifiée de manière

re unique dans la base de connaissances par la combinaison nom/version,

- une fonction d'analyse du composant exécutée sur la plate-forme UNIX analysée qui possède deux paramètres d'entrée, le nom et la version du composant, et qui retourne une valeur nulle lorsque le composant est situé sur la machine ou une valeur différente de zéro dans le cas contraire,
- un commentaire extérieur lié à la classe de composants, l'utilisateur de l'éditeur ayant accès à ce commentaire, cette information étant optionnelle,
- un commentaire interne lié à la classe de composants, seul l'utilisateur du concepteur ayant accès à ce commentaire qui est interdit à l'utilisateur de l'éditeur, cette information étant optionnelle,
- un icône graphique représentant la classe de composants, cette information étant optionnelle.

Un groupe fonctionnel est entièrement défini par les données suivantes:

- un nom de groupe fonctionnel identifiant de manière unique le groupe fonctionnel dans la base de connaissances, cette information étant obligatoire,
- une donnée précisant si le groupe fonctionnel est scalaire ou tabulaire (par défaut, le groupe fonctionnel étant scalaire), cette information étant optionnelle,
- une donnée précisant si le groupe fonctionnel est obligatoire ou non dans le composant, une instance du groupe fonctionnel obligatoire étant créée lorsque le composant est ajouté à un système par l'intermédiaire de l'éditeur,
- un commentaire extérieur lié au groupe fonctionnel, l'utilisateur de l'éditeur ayant accès à ce commentaire, cette information étant optionnelle,
- un commentaire interne lié au groupe fonctionnel, seul l'utilisateur du concepteur ayant accès à ce commentaire qui est interdit à l'utilisateur de l'éditeur, cette information étant optionnelle,
- un icône graphique représentant le groupe fonctionnel, cette information étant optionnelle.

Un attribut est entièrement défini par les données suivantes:

- le nom interne de l'attribut, unique relativement aux autres noms d'attributs,

- la syntaxe de l'attribut qui peut être choisie parmi une liste de valeurs, cette information étant obligatoire,

- 5 - un commentaire interne lié à l'attribut, seul l'utilisateur du concepteur ayant accès à ce commentaire qui est interdit à l'utilisateur de l'éditeur, cette information étant optionnelle,
- 10 - dépendant de la syntaxe, la spécification des vérifications sur la valeur entrée par l'utilisateur pour cet attribut: pour un entier une valeur minimum et/ou une valeur maximum, pour une chaîne de caractères le nombre de caractères exact ou minimum et/ou maximum, pour une liste énumérative un contrôle implicite étant exécuté et les valeurs extérieures à la liste de valeurs permises étant rejetées.
- 15

L'instrumentation de composant est une interface bas-niveau spécifique de la classe de composants et utilisée pour fixer (fonction de type "set") ou lire (fonction de type "get") une information de configuration relative à un quelconque composant de cette classe. L'instrumentation de composant est mise en oeuvre par le fournisseur de composant, la définition de l'instrumentation de composant étant enregistrée dans la base de connaissances et directement liée à la classe de composants. L'instrumentation de composant pour une classe de composants est construite à partir d'un ensemble d'instrumentations appelé instrumentations de groupes fonctionnels et défini pour les groupes fonctionnels constituant cette classe de composants. Par conséquent, définir une instrumentation de composant correspond à définir une instrumentation pour chacun des groupes fonctionnels constituant la classe de composants et la mise en oeuvre d'une instrumentation de composant est donc dépendante du type de systèmes configurés avec le système de configuration. L'instrumentation d'un groupe fonctionnel peut consister soit d'un ensemble de commandes de configuration, soit de quelques fichiers de configuration et ceci, indépendamment du type de plates-formes qui sont configurées. L'instrumentation d'un quelconque groupe fonctionnel qu'il est désiré intégrer au système de configuration doit suivre un ensemble de règles, ces règles sont détaillées ci-dessous pour chaque type d'instrumentation (commande/fichier). L'instrumentation de groupes fonctionnels permet d'exécuter les fonctionnalités suivantes:

- 50 - exécuter une commande préparatoire,
- exécuter une commande postérieure,
- 55 - lire la valeur d'un ensemble d'attributs appartenant au groupe fonctionnel,
- fixer la valeur d'un ensemble d'attributs appartenant au groupe fonctionnel,

- pour un groupe fonctionnel tabulaire, créer une instance de ce groupe fonctionnel,
- pour un groupe fonctionnel tabulaire, lire une liste de toutes les instances de ce groupe fonctionnel,
- pour un groupe fonctionnel tabulaire, supprimer une ou plusieurs instances de ce groupe fonctionnel.

En ce qui concerne la configuration de plates-formes UNIX, dans certains cas, les fonctionnalités détaillées ci-avant peuvent être remplies par des commandes de configuration, en particulier, il est possible de considérer les commandes sous l'interface de SMIT qui permettent d'exécuter des tâches de configuration sur une quelconque plate-forme supportant le système d'exploitation AIX. Le système de configuration selon l'invention qui permet, bien sûr, d'utiliser de telles commandes mais de manière plus efficiente, propose en outre des règles sur les commandes définissant l'instrumentation d'un groupe fonctionnel telles que les suivantes:

- ces commandes ne sont pas interactives,
- elles permettent de correspondre et de s'harmoniser au concept SMIT, l'utilisateur du concepteur définissant pour cela, au sein des composants, des groupes fonctionnels pour autoriser la cohérence entre les commandes de configuration et ledit concept,
- la commande permettant de lire la liste des noms d'instances de groupes fonctionnels renvoie en retour ces noms de telle manière que, les noms retournés sont utilisés comme paramètres d'entrée de la commande autorisant l'exécution de la lecture et qu'il n'existe exactement qu'un seul nom d'instance de groupe fonctionnel par ligne,
- la commande permettant de lire les valeurs d'attributs pour une instance de groupe fonctionnel renvoie en retour ces valeurs selon le même format que celui de SMIT en utilisant un caractère séparateur, l'ordre de retour étant le même que l'ordre des attributs dans le groupe fonctionnel et les valeurs retournées étant montrées comme tel par l'éditeur.

Le système de configuration est conçu pour autoriser également l'intégration de composants configurés en utilisant des fichiers de configuration. Pour cela, les règles concernant l'instrumentation de fichiers sont les suivantes:

- les fichiers de configuration sont des fichiers ASCII,
- chaque groupe fonctionnel doit être instrumenté par

un et un seul fichier de configuration,

- le plus petit élément d'un quelconque fichier de configuration instrumentant une instance d'un groupe fonctionnel est la ligne, ce qui signifie qu'une instance de groupe fonctionnel est représentée par au moins une ligne dans le fichier de configuration lié à ce groupe fonctionnel.

Ainsi, en concordance avec les règles précédentes, une instrumentation d'un groupe fonctionnel pour des plates-formes UNIX utilisant des fichiers de configuration est intégralement définie par (et donc stockée dans la base de connaissances):

- un nom identifiant de manière unique l'instrumentation,
- le gabarit d'instrumentation lié au fichier de configuration pour le groupe fonctionnel et spécifié par l'utilisateur du concepteur, la syntaxe de ce gabarit prenant en compte les groupes fonctionnels tabulaires,
- le nom intégral du chemin menant au répertoire où doit être lu le fichier de configuration,
- le nom intégral du chemin menant au répertoire où doit être écrit le fichier de configuration,
- le mode écriture pour le fichier de configuration: "append" si le fichier de configuration est ajouté à un fichier déjà existant sur la plate-forme configurée, ou "overwrite" si le fichier de configuration remplace un fichier déjà existant sur la plate-forme configurée.

Il y a toujours une feuille d'informations associée au groupe fonctionnel lorsque ce groupe fonctionnel est configuré en utilisant des fichiers de configuration. L'instrumentation n'a aucune influence sur la préparation de la configuration effectuée sur la station de travail, mais a une influence principalement sur les procédés appliqués pour la construction de la configuration désirée d'une plate-forme, c'est-à-dire sur les sous-procédés d'actualisation de la configuration et d'analyse et de contrôle de la configuration.

Le format utilisé doit correspondre au format des données de configuration introduites, d'un côté un quelconque élément de données de configuration est vu selon le format déterminé par l'utilisateur de l'éditeur, alors que le même élément de données de configuration est vu selon un autre format, appelé format produit, du côté interpréteur. Ce choix est régi par les instrumentations, que ces dernières utilisent des commandes ou des fichiers de configuration.

De plus, les données de configuration doivent spécifier le type de liaisons manipulées par l'utilisateur de

l'éditeur. les différents types de liaisons sont intégralement définies par les données suivantes:

- un nom identifiant de manière unique le type de liaison, 5
- les caractéristiques de cette liaison, liaison point à point, pour la connexion d'un système à un autre système ou d'un système à un lien fédératif, liaison fédérative, pour l'interconnexion d'un ensemble de systèmes par l'intermédiaire de liaisons point à point, 10
- un icône graphique représentant le type de liaison, cette information étant optionnelle, 15
- un commentaire extérieur lié au type de liaison, l'utilisateur de l'éditeur ayant accès à ce commentaire, cette information étant optionnelle, 20
- un commentaire interne lié au type de liaison, seul l'utilisateur du concepteur ayant accès à ce commentaire qui est interdit à l'utilisateur de l'éditeur, cette information étant optionnelle. 25

Ces données doivent aussi spécifier un ensemble de composants avec leurs numéros de versions fixés de manière adéquate, spécifiques d'une classe de plates-formes également définie de manière adéquate, elles se présentent de la manière suivante:

- un nom les identifiant de manière unique, 30
- un nom identifiant la classe de plates-formes pour laquelle elles sont destinées à servir, 35
- la liste de couples nom/numéro de version identifiant chacun des composants concernés par ces données, 40
- un commentaire extérieur lié à cet ensemble de composants, l'utilisateur de l'éditeur ayant accès à ce commentaire, cette information étant optionnelle, 45
- un commentaire interne lié à cet ensemble de composants, seul l'utilisateur du concepteur ayant accès à ce commentaire qui est interdit à l'utilisateur de l'éditeur, cette information étant optionnelle. 50

Ce concept d'ensemble de composants est destiné à accélérer la spécification, par l'utilisateur de l'éditeur, de la présence d'un ensemble de produits sur une plate-forme, après quoi, chaque composant d'un quelconque ensemble de composants est manipulé dans son intégralité.

Ces données doivent aussi spécifier une liste de composants et sont donc utilisées pour définir, en ter-

mes de configuration, la dépendance d'un composant relativement à un ensemble de composants. Une telle dépendance en termes de configuration est identifiée par:

- le nom identifiant cette dépendance de manière unique, 5
- le nom identifiant toute classe de composants appartenant à cette dépendance de configuration, 10
- l'ensemble de numéros de version lié à cette classe de composants. 15

Une classe de composants est liée à une dépendance de configuration par la relation composant/dépendance.

La relation plate-forme /composant, quant à elle, permet d'obtenir la liste de tous les composants, selon la combinaison nom/version, qui peuvent être situés sur n'importe quelle plate-forme d'une classe donnée. Cette relation est intégralement définie par:

- le nom identifiant la classe de plates-formes, 25
- le nom de la classe de composants en relation avec cette classe de plates-formes, 30
- le numéro de version de la classe de composants en relation avec cette classe de plates-formes. 35

La relation entre une classe de plates-formes et une classe de composants est définie seulement si ces deux classes sont définies dans la base de connaissances.

Les données doivent aussi spécifier une relation composant/dépendance entre une classe de composants et une liste de classes de composants dans la base de connaissances. Cette relation exprime d'une part, les conditions préalables à remplir, relativement à d'autres composants, pour qu'un composant puisse être configuré, certains composants requérant la présence d'autres composants pour être configuré et d'autre part, la compatibilité qu'un composant doit présenter, relativement à d'autres composants. Cette relation est intégralement définie par:

- le nom identifiant la classe de composants déterminée, 45
- le nom identifiant la dépendance de configuration liée au nom de la classe de composants déterminée. 50

Tout attribut est destiné à définir une partie d'une information de configuration, par conséquent, un attribut est lié à un grand nombre de groupes fonctionnels d'un même composant ou de composants différents. Réciproquement, un groupe fonctionnel est lié à un grand

nombre d'attributs. Des données sont donc utilisées pour définir cette relation attribut/groupe fonctionnel, ces données sont les suivantes:

- le nom interne d'un attribut l'identifiant de manière unique,
- le nom identifiant le groupe fonctionnel lié à l'attribut,
- le nom de l'attribut dans le groupe fonctionnel, appelé nom externe de l'attribut, c'est le nom qui est affiché pour l'utilisateur de l'éditeur,
- un indicateur ou drapeau spécifiant si l'attribut est obligatoire dans le groupe fonctionnel, c'est-à-dire si une valeur de cet attribut doit être fournie lorsqu'une instance de ce groupe fonctionnel est configurée,
- un commentaire, typiquement un message d'aide, lié à l'attribut lorsque cet attribut est sollicité dans ce groupe fonctionnel, l'utilisateur de l'éditeur pouvant y accéder pour obtenir une information sur cet attribut dans le contexte d'un groupe fonctionnel, cette information étant optionnelle, lorsqu'elle n'est pas présente, le commentaire pour l'attribut étant utilisé,
- une valeur par défaut associée à l'attribut dans le groupe fonctionnel.

Il est ici à noter qu'une instance de groupe fonctionnel est nommée par la valeur du premier attribut défini pour ce groupe fonctionnel.

Une instrumentation se réfère toujours à un groupe fonctionnel d'un composant, cette relation est destinée à lier une instrumentation à un groupe fonctionnel appartenant à une classe de composants. Par conséquent, la relation entre une instrumentation d'un groupe fonctionnel et ce groupe fonctionnel est décrite par les données suivantes:

- le nom de la classe de composants intégrant le groupe fonctionnel,
- le numéro de version de cette classe de composants,
- le nom identifiant l'instrumentation,
- le nom identifiant le groupe fonctionnel lié à cette instrumentation.

Un composant est constitué d'un ensemble de groupes fonctionnels, cette relation lie un groupe fonctionnel à un composant, ce qui signifie que le groupe fonctionnel est contenu dans le composant. Par consé-

quent, la relation entre un groupe fonctionnel et le composant auquel il appartient est décrite par données suivantes:

- le nom identifiant le groupe fonctionnel,
- le nom de la classe de composants intégrant le groupe fonctionnel,
- le numéro de version de cette classe de composants.

Une fois ces précisions apportées relativement à la base de connaissances, il est possible de mieux appréhender l'objet des différents sous-procédés appliqués selon l'invention. Ainsi, le sous-procédé d'édition de la base de connaissances a pour but de fournir à l'utilisateur d'une station de travail de type PC un accès à la base de connaissances lui permettant de créer cette dernière à partir de rien, de l'éditer, de la modifier et de l'enregistrer.

Comme la base de connaissances est liée à une station de travail, l'utilisateur doit pouvoir conserver l'intégrité d'une station de travail à une autre station de travail afin d'éviter toute redondance en ce qui concerne les noms et les versions.

Ce sous-procédé d'édition de la base de connaissances utilise comme données d'entrée les données de la base de connaissances enregistrées dans la station de travail ainsi que les données de l'utilisateur destinées à modifier le contenu de la base de connaissances. Les données de l'utilisateur sont entrées au travers de l'interface utilisateur graphique du concepteur qui est l'interface d'entrée pour le flux de données ainsi que l'interface de contrôle. Ce sous-procédé utilise les données de la base de connaissances également comme données de sortie. L'ensemble des fonctionnalités du concepteur propres à l'interface utilisateur et aux fonctions du "noyau" ainsi que les données de l'utilisateur relatives à la configuration désirée sont utilisées. L'interface d'entrée pour le sous-procédé d'édition de la base de connaissances, en ce qui concerne les données d'entrée relatives à la base de connaissances est, par la suite et pour des besoins de simplification et de concision, appelée interface de connaissances. De cette manière, lors d'une opération de création, le traitement propre aux tâches d'initialisation de la base de connaissances est exécuté, alors que les données de sortie qui à ce moment n'existent pas correspondent à une base de connaissances vide. Egalement, lors d'une opération d'édition, les données de la base de connaissances sont prises comme données d'entrée et les données fournies en sortie sont les données de la base de connaissances éditées au travers de l'interface utilisateur graphique du concepteur. De même, lors d'une opération de modification ou suppression les données de la base de connaissances sont prises comme données d'entrée ainsi que les données de configuration désirée entrées par

l'utilisateur alors que les données offertes en sortie sont les données de la base de connaissances modifiée. Enfin, lors d'une opération d'enregistrement il permet de sauvegarder la base de connaissances en enregistrant les données de ladite base de connaissances. Ce sous-procédé est entièrement exécuté par le concepteur du système de configuration et il est important ici de noter que le concepteur peut manipuler plusieurs bases de connaissances, aussi dans ce cas, l'éditeur peut lier une configuration à une base de connaissances.

Le sous-procédé de détermination d'une configuration permet de construire une configuration désirée à partir de l'éditeur du système de configuration. Une configuration pour ledit éditeur correspond à un ensemble hétérogène de plates-formes qui sont interconnectées au moyen de liaisons, cette configuration étant entrée au moyen de feuilles d'informations permettant d'identifier les composants associés à ces plates-formes, sachant qu'une configuration donnée est toujours liée à la base de connaissances.

Hauptprozess
La détermination d'une configuration est l'opération de l'utilisateur qui consiste à construire par étapes, à partir d'une configuration existante ou tout simplement d'aucune configuration, une configuration déterminée. Le sous-procédé lui fournissant les moyens pour déterminer cette configuration. Pour cela, dans une première étape, l'utilisateur, de manière optionnelle, sélectionne une configuration qui est, de manière unique identifiée par un nom et un numéro de version, de préférence dans le format VRMF. Il est rappelé que, comme la configuration est exécutée sur une station de travail, l'utilisateur doit pouvoir conserver l'intégrité des configurations déterminées d'une station de travail à une autre station de travail. Dans une seconde étape, il construit ou modifie à l'écran la topologie du réseau, c'est-à-dire spécifie les plates-formes et les connecte au réseau au moyen de liaisons. Puis dans une troisième étape, sélectionne une plate-forme, spécifie les composants qui sont installés sur cette plate-forme, définit les instances des groupes fonctionnels et, pour chaque instance entre des informations de configuration dans les feuilles d'informations adéquates. Cette opération est répétée pour chaque plate-forme que l'utilisateur désire configurer ou associer dans la configuration d'ensemble. Enfin dans une quatrième étape, il réalise la configuration en engendrant les fichiers de configuration de type MIF qui sont distribués sur chaque plate-forme configurée.

Subprozess
Pour préciser, l'interface de contrôle est l'interface utilisateur graphique de l'éditeur. L'ensemble des fonctionnalités de l'éditeur propres à l'interface utilisateur et aux fonctions du "noyau", quelques fonctionnalités propres au "noyau" du concepteur pour l'accès à la base de connaissances ainsi que les données de l'utilisateur relatives à la configuration désirée sont utilisées. Ainsi, les données d'entrée utiles à l'application du sous-procédé de détermination d'une configuration sont les suivantes:

- la base de connaissances enregistrée sur la station de travail de type PC, l'interface permettant d'accéder à cette base est l'interface de connaissances,
- 5 - les données de topologie et les données de configuration qui sont entrées par l'utilisateur par l'intermédiaire de l'interface utilisateur graphique de l'éditeur,
- 10 - le nom de la configuration, si une configuration déjà existante est éditée, cette donnée étant optionnelle.

De cette manière, lors d'une opération de création, le traitement relatif à l'initialisation de toute configuration est exécuté, alors que les données de sortie, qui à ce moment n'existent pas, correspondent à une configuration vide, une lecture d'informations de la base de connaissances étant nécessaire au travers de l'interface de connaissances. Egalement, lors d'une opération d'édition, le nom de la configuration entré par l'intermédiaire de l'interface utilisateur graphique de l'éditeur ainsi que les données de la base de connaissances par l'intermédiaire de l'interface de connaissances sont pris comme données d'entrée de même que l'ensemble des données correspondant à des configurations déjà existantes sont aussi pris comme données d'entrée desquelles il est extrait la configuration désirée par l'utilisateur, alors que les données fournies en sortie par l'intermédiaire de l'interface utilisateur graphique de l'éditeur correspondent à la configuration désirée par l'utilisateur. De même, lors d'une opération de modification de la topologie et des informations de configuration, les données de configuration entrées par l'utilisateur par l'intermédiaire de l'interface utilisateur graphique de l'éditeur sont prises comme données d'entrée, pour fournir une configuration modifiée qui est éditée au travers de l'interface utilisateur graphique de l'éditeur, l'opération de modification de topologie fournissant la topologie et l'opération de modification des informations de configuration fournissant la topologie configurée. Enfin, lors d'une opération de réalisation de la configuration, la topologie configurée est prise comme données d'entrée et la configuration désirée est éditée par l'intermédiaire de l'interface utilisateur graphique de l'éditeur, ladite configuration étant enregistrée dans un ensemble de fichiers de configuration, de préférence de type MIF, au travers de l'interface d'écriture des fichiers de configuration. Les fichiers de configuration contiennent les informations suivantes:

- les informations de configuration entrées par l'utilisateur,
- la structure des composants qui sont configurés,
- 55 - l'instrumentation.

Dans cet environnement complexe en mode client-

serveur, il peut être précisé à cet endroit, qu'avantageusement l'éditeur fournit, entre autres, un mécanisme mettant en oeuvre la gestion de l'unicité d'un objet et donc de données à travers le réseau représenté. Grâce à ce mécanisme, un utilisateur qui prépare une configuration sur une station de travail du système de configuration, ne doit entrer qu'une et une seule fois les mêmes données de configuration.

Pour cela, ce mécanisme d'unicité de la configuration permet de propager la valeur d'un attribut, lié à une instance d'un groupe fonctionnel, vers d'autres attributs liés à d'autres instances de groupes fonctionnels, que ces instances soient situées sur le même système ou non. Par conséquent, des ensembles d'attributs sont définis dans une configuration donnée en partageant la même valeur, et chaque fois qu'un attribut est modifié par l'utilisateur, tous les attributs appartenant à un même ensemble sont automatiquement modifiés. Ainsi, comme chaque attribut est identifié, de manière interne et unique, par un nom interne, les attributs communs à différents groupes fonctionnels auront le même nom interne, c'est cette caractéristique qui est utilisée pour mettre en oeuvre l'unicité de la configuration. Les informations suivantes qui sont liées à un attribut, permettent d'identifier à quel groupe fonctionnel il appartient:

- au regard du système, le nom du système sur lequel le groupe fonctionnel considéré est situé,
- au regard de la classe de composants, le nom de la classe de composants renfermant le groupe fonctionnel considéré,
- au regard de l'identifiant de l'instance de groupe fonctionnel, le nom de l'instance du groupe fonctionnel considéré,
- au regard du nom externe, le nom externe de l'attribut dans le groupe fonctionnel considéré.

La partie d'information concernant l'identifiant de l'instance de groupe fonctionnel doit être spécifiée puisque plusieurs instances du même groupe fonctionnel partagent le même identifiant d'attribut. La partie d'information concernant le système doit être spécifiée lorsqu'une valeur d'attribut propre à un système est distribuée sur un autre système. L'unicité de la configuration est rendue effective par l'éditeur au moment où l'utilisateur remplit les feuilles d'informations avec les données de configuration. Il est à noter que cette unicité ne doit pas être limitée par le format des données d'entrée. En effet, pour l'utilisateur de l'éditeur les mêmes données de configuration entrées pour différents groupes fonctionnels doivent avoir le même format, mais du point de vue de l'instrumentation, les mêmes données de configuration entrées pour configurer différents groupes fonctionnels peuvent avoir un format différent. Ce problème de format est avantageusement résolu par l'in-

frastructure de communications du système de configuration et plus particulièrement grâce aux instrumentations. De fait et comme cela a été précédemment explicité, le format utilisé doit correspondre au format des données de configuration introduites, d'un côté un quelconque élément de données de configuration est vu selon le format déterminé par l'utilisateur de l'éditeur, alors que le même élément de données de configuration est vu selon un autre format, appelé format produit, du côté interpréteur. Ce choix est régi par les instrumentations, que ces dernières utilisent des commandes ou des fichiers de configuration.

Le sous-procédé d'actualisation de la configuration désirée permet de réaliser de manière effective ladite configuration sur la station à configurer adéquate à partir de la configuration préparée par le sous-procédé de détermination de la configuration. Cette fonctionnalité du système de configuration est mise en oeuvre par l'interpréteur de la station UNIX à configurer. Ce sous-procédé a pour objet de convertir la réalisation abstraite de la configuration obtenue par application du sous-procédé de détermination de la configuration en une configuration réelle d'une station désirée configurée. Après application de ce sous-procédé, ladite station est configurée de manière nouvelle, comme préparée, et ceci à partir, lorsque cela est désiré, d'une quelconque configuration antérieure. Ce sous-procédé permet de prendre en compte tout type de configuration, une première configuration (pas de configuration antérieure), une reconfiguration qui est une nouvelle configuration obtenue à partir d'une configuration antérieure, ou une déconfiguration c'est-à-dire une configuration antérieure reprise en ayant supprimé certains éléments de cette configuration.

L'interface de contrôle de ce sous-procédé est en fait l'interpréteur de la station à configurer. Pour l'application de ce sous-procédé, les différents éléments utilisés ou à prendre en considération sont les suivants:

- l'utilisateur qui est responsable du lancement du sous-procédé, sachant qu'une fois lancé, ce sous-procédé ne requiert plus d'intervention humaine,
- l'interpréteur qui met en oeuvre ce sous-procédé,
- éventuellement l'outil SMIT sur une plate-forme AIX qui permet aussi de lancer ce sous-procédé qui peut, en outre, être lancé à partir d'un interpréteur de commandes (appelé par l'homme du métier "shell") par l'intermédiaire d'une interface de ligne de commande (CLI) pour des plates-formes UNIX (AIX ou non).

Les données d'entrée et l'interface de ce sous-procédé sont constituées des éléments suivants:

- les fichiers de configuration, de préférence de type MIF, décrivant d'une part, la configuration que le

sous-procédé installe et d'autre part les éléments de la base de connaissances utilisés par l'interpréteur, les fichiers de configuration étant accédés au travers d'une interface spécifique à l'interpréteur,

- la configuration effective du produit, c'est-à-dire toutes les données de configuration pour tous les produits de la station, cette configuration étant conservée et accessible dans la station dans des fichiers de configuration spécifiques, l'interface permettant d'accéder à ces données d'entrée étant constituée d'un ensemble de commandes spécifiques,
- le nom de la station à configurer entré par l'utilisateur. ce nom correspond au nom du système spécifié dans la configuration préparée par le sous-procédé de détermination de la configuration.

Les données de sortie et l'interface de ce sous-procédé sont constituées des éléments suivants:

- la nouvelle configuration effective du produit qui est, comme la précédente, conservée et accessible dans la station dans des fichiers de configuration spécifiques,
- les enregistrements de la configuration effective, c'est-à-dire les enregistrements contenant les informations relatives à l'analyse des tâches de configuration effectivement exécutées comportant également l'ensemble des lignes et des résultats de commandes.

Lors de l'application de ce sous-procédé, trois opérations sont successivement effectuées, une opération de vérification de la configuration réellement installée, une opération d'analyse de la configuration effective relativement aux contenus des fichiers de configuration et enfin une opération d'activation de la configuration désirée.

L'opération de vérification de la configuration consiste à effectuer les vérifications relativement à ce qui est réellement installé sur la station, pour cela, sont utilisées des fonctions d'analyse décrites dans les fichiers de configuration transmis pour vérifier la totale cohérence entre la configuration effective de la station et les informations contenues dans les fichiers de configuration, de préférence de type MIF, en termes de contenus de produit et de numéros de version. Par conséquent, les données d'entrée sont ici le nom du système ainsi que les éléments de la base de connaissances contenus dans les fichiers de configuration.

L'opération d'analyse de la configuration effective consiste à comparer la configuration effective de la station aux contenus des fichiers de configuration en générant les actions appropriées. Pour cela, pour chaque composant dont la description est donnée dans les fi-

chiers de configuration, les contenus desdits fichiers de configuration sont comparés à la configuration effective. Egalement, pour chaque groupe fonctionnel de chaque composant listé dans les fichiers de configuration les instances réellement configurées sont déterminées, si les instances sont configurées mais ne sont pas présentes dans les fichiers de configuration, elles sont supprimées, si les instances sont présentes dans les fichiers de configuration mais ne sont pas configurées, elles sont créées, si les instances sont à la fois configurées et présentes dans les fichiers de configuration, elles sont modifiées. Puis, lorsque c'est nécessaire, c'est-à-dire pour une création ou une modification, les mots-clés spécifiant les instrumentations sont remplacés par les valeurs trouvées dans les fichiers de configuration. Enfin, il est déterminé l'ordre suivant lequel la configuration des groupes fonctionnels dans un composant et la configuration des composants sont effectuées.

L'opération d'activation de la configuration désirée consiste à lancer la configuration à partir des données d'entrée constituées par les informations résultant de l'opération d'analyse précédente. Lancer la configuration signifie installer les fichiers de configuration à la bonne place sur la station configurée et exécuter les commandes de configuration récupérées lors de l'opération d'analyse précédente.

Le sous-procédé d'analyse et de contrôle de la configuration sur la station configurée permet de récupérer la configuration de cette station et de fournir les informations de configuration dans un format aisément accessible au sous-procédé d'exploration de la configuration d'une station configurée à partir d'une station de travail. Cette fonctionnalité du système de configuration est mise en oeuvre par l'interpréteur de la station UNIX configurée. L'interface de contrôle de ce sous-procédé est donc l'interpréteur de la station configurée. Pour l'application de ce sous-procédé, les différents éléments utilisés ou à prendre en considération sont les suivants:

- l'utilisateur qui est responsable du lancement du sous-procédé, sachant qu'une fois lancé, ce sous-procédé ne requiert plus d'intervention humaine,
- l'interpréteur qui met en oeuvre ce sous-procédé sur une station configurée déterminée,
- éventuellement l'outil SMIT sur une plate-forme AIX qui permet aussi d'appeler ce sous-procédé qui peut, en outre, être lancé à partir d'un interpréteur de commandes ("shell") par l'intermédiaire d'une interface de ligne de commande (CLI) pour des plates-formes UNIX (AIX ou non).

Les données d'entrée et l'interface de ce sous-procédé sont constituées des éléments suivants:

- la partie de la base de connaissances décrivant les stations UNIX, l'accès à cette base étant permis par

l'intermédiaire d'une interface spécifique à l'inter-préteur qui est la même que l'interface utilisée par le sous-procédé d'actualisation de la configuration désirée, cette partie de la base de connaissances représentée dans les fichiers de configuration, de préférence de type MIF, comprenant: la liste de tous les composants (noms et versions) contenus dans la base de connaissances incluant la fonction d'analyse des composants, la description des groupes fonctionnels concernés ainsi que les instrumentations liées aux groupes fonctionnels,

- la configuration effective du produit, c'est-à-dire les données de configuration pour tous les produits de la station, cette configuration étant conservée et accessible dans la station dans des fichiers de configuration spécifiques, l'interface permettant d'accéder à ces données d'entrée étant constituée d'un ensemble de commandes spécifiques également appelées instrumentations de composants, les instrumentations de composants étant récupérées dans la partie de la base de connaissances déterminée pour le présent sous-procédé.

Les données de sortie et l'interface de ce sous-procédé sont constituées des fichiers de configuration, de préférence de type MIF, décrivant la configuration de la station analysée, ces fichiers de configuration contenant les informations de configuration des composants installés sur la station.

Lors de l'application de ce sous-procédé, trois opérations sont successivement effectuées, une opération d'analyse de la station, une opération de lecture des données de configuration et enfin d'une opération de création des fichiers de configuration.

L'opération d'analyse de la station consiste à déterminer la liste des composants installés sur la station analysée et récupérer le nom et la version de chaque composant à l'aide de la fonction d'analyse récupérée dans la base de connaissances. Par conséquent, les données d'entrée sont ici les données de la base de connaissances contenues dans les fichiers de configuration et les données de sortie sont la liste des composants.

L'opération de lecture des données de configuration consiste à récupérer lesdites données de configuration propres aux produits installés sur la station, les données d'entrée sont donc:

- les données de la base de connaissances dans lesquelles, pour chaque composant sont extraits: d'une part pour les groupes fonctionnels dont la configuration est mise en oeuvre par des commandes, les commandes permettant de lire la liste des instances des groupes fonctionnels constituant le composant ainsi que pour chaque instance de groupe fonctionnel, la liste des valeurs d'attributs et d'autre part pour les groupes fonctionnels dont la configuration est mise en oeuvre par les fichiers de

configuration, lesdits fichiers de configuration,

- la liste des composants situés sur la station analysée fournie lors de l'opération d'analyse de la station.

Les données de sortie fournies pour effectuer l'opération de création des fichiers de configuration sont alors les données de configuration des produits présentées dans un format interne.

L'opération de création des fichiers de configuration consiste à prendre, comme données d'entrée, les données de sortie générées lors de l'opération de lecture des données de configuration et à exécuter les tâches suivantes:

- pour chaque composant, construire l'ensemble des fichiers de configuration, de préférence de type MIF, contenant les données de configuration dudit composant dans un format déterminé,
- pour l'ensemble des composants, construire l'arbre de tous les fichiers de configuration dans le format déterminé,
- enregistrer cet arbre à un emplacement spécifié par l'utilisateur.

Le sous-procédé d'exploration de la configuration d'une station configurée à partir d'une station de travail permet d'éditer, sur l'éditeur de cette station de travail, une configuration d'une quelconque station configurée et analysée. Ce sous-procédé permet d'insérer la configuration de la station analysée dans une configuration éditée sur la station de travail, insérer signifiant ici:

- afficher la configuration de la station analysée dans le dessin de la configuration,
- relier la configuration de la station analysée dans la représentation du réseau, étant donné les données de configuration de ce système,
- fournir l'édition des données de configuration récupérées à l'interface utilisateur graphique.

Ce sous-procédé qui a des caractéristiques en commun avec l'opération d'édition effectuée lors de l'application du sous-procédé de détermination de la configuration est donc mis en oeuvre par l'éditeur. L'interface de contrôle de ce sous-procédé est ainsi l'interface utilisateur graphique de l'éditeur, de même, l'ensemble des fonctionnalités de l'éditeur propres à l'interface utilisateur et aux fonctions du "noyau", quelques fonctionnalités propres au "noyau" du concepteur pour l'accès à la base de connaissances ainsi que les données de l'utilisateur relatives à la configuration désirée sont utilisées. Ainsi, les données d'entrée utiles à l'application

de ce sous-procédé d'exploration sont les suivantes:

- la base de connaissances enregistrée sur la station de travail de type PC, l'interface permettant d'accéder à cette base étant l'interface de connaissances,
- les fichiers de configuration, de préférence de type MIF, générés par le sous-procédé d'analyse et de contrôle de la configuration sur la station configurée, lus au travers de l'interface de lecture des fichiers de configuration,
- les structures de données spécifiant la configuration éditée dans laquelle la configuration de la station analysée est insérée.

Les données de sortie fournies par le sous-procédé d'exploration spécifient la configuration éditée dans laquelle la configuration de la station analysée a été ajoutée.

Lors de l'application de ce sous-procédé, trois opérations sont successivement effectuées, une opération de lecture des données de configuration, une opération de connexion dans le réseau représenté et enfin une opération d'affichage.

L'opération de lecture des données de configuration consiste à, d'une part, vérifier que la configuration est adaptée à la base de connaissances et d'autre part, remplir les structures de données internes avec les contenus des fichiers de configuration, ces structures de données représentant la configuration de la station analysée étant utilisées pour effectuer les opérations subséquentes de connexion et d'affichage. Pour cela, les données d'entrée utilisées sont les données de la base de connaissances et les fichiers de configuration accédés au travers de l'interface de lecture des fichiers de configuration.

L'opération de connexion dans le réseau représenté consiste à examiner la configuration éditée ainsi que les données de configuration de la station analysée et à déterminer les liaisons permettant de connecter la station analysée aux autres stations présentes dans la configuration éditée. Pour cela, les données d'entrée utilisées sont les structures de données représentant la configuration éditée et les structures de données représentant les données de configuration de la station analysée. Cette opération de connexion modifie les structures de données représentant la configuration éditée, lorsque les liaisons sont effectuées.

L'opération d'affichage consiste à dessiner la nouvelle topologie après insertion de la station analysée en incluant en premier lieu, la représentation graphique de la station analysée, en second lieu, la représentation graphique des liaisons permettant de connecter la station analysée au réseau, en troisième lieu, la représentation des données de configuration de la station analysée dans les feuilles d'informations. Pour cela, les données d'entrée utilisées sont les structures de don-

nées représentant la configuration éditée, modifiées par l'opération de connexion précédente.

Les principales interactions entre ces différents sous-procédés sont ci-après explicitées. Une première interaction ainsi existe, lors du passage du sous-procédé de détermination de la configuration au sous-procédé d'actualisation de la configuration désirée pour l'exécution de la préparation de la configuration, de sa distribution et de sa détermination effective. Les fichiers de configuration sont générés par le sous-procédé de détermination de la configuration, la totalité des fichiers de configuration est alors transférée de la station de travail vers la station à configurer dans laquelle est mis en oeuvre le sous-procédé d'actualisation de la configuration désirée, soit au moyen de disquettes, soit en utilisant des moyens électroniques de transfert de fichiers. Ces fichiers sont des fichiers ASCII et le même ensemble de fichiers est transféré vers toutes les stations concernées par la configuration désirée. Avec une application UNIX, lorsqu'un transfert de fichiers est réalisé par l'intermédiaire d'une disquette, les fichiers sont copiés sur chacune des stations concernées par le sous-procédé d'actualisation de la configuration désirée en utilisant une commande UNIX (dosread). Les fichiers de configuration de type MIF contiennent toutes les informations nécessaires à la configuration des stations concernées par cette configuration, le contenu de ces fichiers, pour tous les composants à configurer, est le suivant:

- la description des groupes fonctionnels liés,
- les instrumentations liées aux groupes fonctionnels,
- les données de configuration, pour les groupes fonctionnels configurés,
- l'ordre de configuration entre les groupes fonctionnels et les composants, cet ordre étant extrait de la base de connaissances.

Une seconde interaction est décrite relativement au passage du sous-procédé d'analyse et de contrôle de la configuration vers le sous-procédé d'exploration de la configuration d'une station configurée à partir d'une station de travail pour exécuter l'analyse de la configuration. Les fichiers de configuration de type MIF contenant les données de la base de connaissances sont nécessaires à l'application du sous-procédé d'analyse et de contrôle de la configuration. C'est à l'utilisateur de fournir ces fichiers et de spécifier leur emplacement sur les stations. De plus, l'éditeur donne à l'utilisateur la possibilité de produire les fichiers de configuration de type MIF contenant les données de la base de connaissances, pour une quelconque station, et de les insérer sur une disquette ou autre. Ces fichiers de configuration de type MIF contiennent toutes les informations nécessaires

res pour extraire la configuration effective d'une quelconque station, le contenu de ces fichiers est le suivant:

- la liste de tous les composants contenus dans la base de connaissances, incluant la fonction d'analyse des composants,
- la description des groupes fonctionnels liés,
- les instrumentations liées aux groupes fonctionnels.

Le sous-procédé d'analyse et de contrôle de la configuration génère les fichiers de configuration de type MIF représentant la configuration de la station analysée et les range dans le répertoire spécifié par l'utilisateur. La structure de ces fichiers de configuration de type MIF est la même que celle générée par l'éditeur. La totalité des fichiers est transférée de la station analysée par le sous-procédé d'analyse et de contrôle de la configuration vers la station de travail en utilisant soit des disquettes, soit des moyens électroniques de transfert de fichiers. Avec une application UNIX, lorsqu'un transfert de fichiers est réalisé par l'intermédiaire d'une disquette, les fichiers sont copiés sur les disquettes par le sous-procédé d'actualisation de la configuration désirée en utilisant une commande UNIX (doswrite).

Une troisième interaction existe lors du passage du sous-procédé d'édition de la base de connaissances vers les divers sous-procédés qui nécessitent des accès à la base de connaissances manipulée par ledit sous-procédé d'édition, les interactions étant effectuées au travers de l'interface de connaissances.

Pour conclure, le système de configuration de logiciels préconfigurés sur des systèmes ouverts et le procédé mis en oeuvre conformément à l'invention permettent de réaliser une configuration automatique de logiciels sur des serveurs ou machines travaillant dans des environnements complexes en mode client-serveur selon une architecture distribuée et offre un service de configuration global aux utilisateurs désirant déployer rapidement un grand nombre de systèmes à partir d'un seul point. A partir d'un seul point, d'une même console d'une station de travail, il est possible de préparer une configuration de référence qui peut être dupliquée sur différentes stations, ou de configurer de manière spécifique une quelconque station, d'un ensemble hétérogène sur le réseau, l'interpréteur, de manière intelligente, à la différence de ce qui se passait dans l'art antérieur, se chargeant de discriminer automatiquement entre les paramètres initiaux de configuration et les paramètres propres à la configuration désirée en choisissant de manière adéquate et automatique ces derniers. La configuration désirée de la station est préparée sur la console d'un microordinateur en utilisant une application conviviale du type "Windows" pour remplir une ou plusieurs "feuilles" d'informations sur l'écran relative à la topologie et aux différents paramètres désirés, et ceci

tout simplement par sélection sur l'écran pour dessiner la topologie du réseau, fixer toute sorte de paramètres et ainsi créer l'image de la configuration souhaitée, diffusée, après génération de fichiers de configuration, vers la machine sur laquelle elle sera chargée pour être exécutée, l'interpréteur permettant ce chargement et faisant avantageusement la distinction entre ce qui existe et ce qui est désiré, alors qu'en outre, cette configuration peut être contrôlée et consultée à partir du microordinateur. L'adaptabilité et l'intelligence de l'interpréteur font qu'il est capable d'identifier la configuration présente et en fonction de la configuration transmise désirée, de reconnaître s'il s'agit d'une première configuration, d'une reconfiguration ou encore d'une déconfiguration pour effectuer alors la commande appropriée. Il est aussi possible de configurer de manière automatique des serveurs similaires pour compléter ou modifier une infrastructure, simplement à partir de moyens de chargement connus. Grâce à ce nouveau concept homogène et centralisé de configuration par commande, l'ouverture à de nouveaux objets configurables est autorisée facilitant tout déploiement ou migration aujourd'hui non permis, y compris le déploiement d'applications d'un client intégrateur de service. Cette dernière caractéristique d'extensibilité est rendue possible du fait que des dialogues de configuration et des commandes spécifiques peuvent être aisément créés et qu'ainsi de nouveaux objets peuvent être ajoutés via le concepteur dont c'est précisément l'objet. Une telle avantageuse convivialité était jusqu'à présent interdite aux réseaux de systèmes hétérogènes en environnement distribué. Parmi les nombreux avantages présentés, il est possible de citer l'extrême simplicité d'utilisation et d'application des procédures de configuration, tandis qu'une meilleure visibilité de la configuration installée est offerte, le graphique de ladite configuration étant directement affichée sur l'écran. Egalement, le temps d'apprentissage pour l'utilisation de telles procédures ainsi donc que le savoir-faire nécessaire à leur exécution peuvent être considérablement réduits et ceci même relativement à un large déploiement de systèmes dupliqués ou non dans des sites aléatoirement et géographiquement distribués. De plus, la productivité et l'efficacité sont très nettement améliorées puisque l'administrateur permet de configurer facilement et rapidement les systèmes en réseau sans oublier que l'automatisation du processus et les possibilités de duplication sont bien supérieures permettant ainsi de réduire les coûts et les délais. Une configuration globale cohérente associée au fait qu'une plus faible technicité est requise pour installer le logiciel implique nécessairement, pour l'administrateur, un meilleur contrôle de la gestion du déploiement des systèmes, ce qui a pour principal effet d'améliorer la fiabilité de l'ensemble. En outre, une très grande autonomie est autorisée à un client utilisateur qui, achetant un logiciel, pourra l'installer lui-même et ceci de manière immédiate, car il n'aura plus besoin systématiquement d'une assistance technique. Enfin, avantageusement,

dans cet environnement complexe en mode client-serveur, l'éditeur fournit des mécanismes puissants et pratiques tels que notamment des mécanismes de gestion de l'unicité d'un objet et donc de données à travers le réseau représenté.

Revendications

1. Système de configuration de logiciels préconfigurés sur des systèmes ouverts en réseau dans un environnement distribué, caractérisé en ce qu'il est constitué, d'une part sur une station de travail, d'un dispositif de conception appelé concepteur, d'un éditeur et d'une base de connaissances, et d'autre part sur une station à configurer d'un interpréteur, le concepteur permettant d'accéder en lecture et écriture à la base de connaissances contenant les données décrivant les différents éléments du réseau qui peuvent être configurés et les liaisons entre ces différents éléments, ledit concepteur permettant de définir les différentes possibilités de configuration de l'éditeur sur lequel est visualisée la configuration désirée, éditeur qui utilise les informations de la base de connaissances relatives à la topologie du réseau et les paramètres de configuration des différents éléments composant le réseau pour construire une entité homogène configurable et chargeable, à l'aide de moyens de chargement, sur la station à configurer dans laquelle l'interpréteur exécute les opérations de configuration en transformant les paramètres de configuration initiaux en des paramètres propres à la configuration désirée également utilisables par l'éditeur, ledit système mettant en oeuvre un procédé de configuration de logiciels préconfigurés.
2. Système de configuration de logiciels préconfigurés selon la revendication 1, caractérisé en ce que le concepteur utilise une interface utilisateur graphique du type "Windows" qui permet de définir de nouvelles classes de plates-formes et de nouvelles classes de composants de systèmes pour étendre les informations de la base de connaissances utilisées par l'éditeur, en offrant pour cela la possibilité de définir, modifier ou supprimer des classes de plates-formes, également de définir, modifier ou supprimer des classes de composants de systèmes et les feuilles d'informations associées à ces classes et enfin relativement aux classes de composants, de définir, modifier ou supprimer des informations des classes de plates-formes relatives à la configuration désirée.
3. Système de configuration de logiciels préconfigurés selon la revendication 1, caractérisé en ce que l'éditeur utilise une interface utilisateur graphique du type "Windows" qui, en premier lieu, fournit des moyens graphiques à l'utilisateur pour construire à l'écran une topologie du réseau représentant l'ensemble des systèmes hétérogènes avec leurs interconnexions, pour chaque système représenté une feuille ou un ensemble de feuilles d'informations de configuration étant rempli, qui également fournit des mécanismes de sauvegarde, d'ouverture et de restauration des configurations définies par l'utilisateur ainsi que des mécanismes de gestion de l'unicité d'un objet à travers le réseau représenté, qui de plus, pour une configuration spécifiée par l'utilisateur, génère un ensemble de fichiers de configuration dans un format déterminé, ces fichiers étant utilisés pour générer une information de configuration qui est exécutée sur la plate-forme désirée configurée et enfin qui, à partir d'un ensemble de fichiers de configuration récupéré sur une plate-forme configurée, extrait la configuration réelle de cette plate-forme.
4. Système de configuration de logiciels préconfigurés selon la revendication 1, caractérisé en ce que l'interpréteur d'une part, lit les fichiers de configuration générés par l'éditeur dans le format déterminé en faisant la différence entre une première configuration, une reconfiguration ou une déconfiguration, puis convertit lesdits fichiers en directives de configuration pour la plate-forme sur laquelle il est situé et enfin effectue la configuration désirée de cette plate-forme et d'autre part, lit la configuration effective de la plate-forme, puis génère les fichiers de configuration dans le format déterminé contenant les informations de configuration des composants de cette plate-forme, lesdits fichiers de configuration étant alors utilisables par l'éditeur.
5. Procédé de configuration de logiciels préconfigurés mis en oeuvre par le système de configuration selon l'ensemble des revendications précédentes, caractérisé en ce qu'il est divisé en une pluralité de sous-procédés, un sous-procédé d'édition de la base de connaissances permettant à un utilisateur d'une station de travail d'accéder à la base de connaissances pour créer, éditer, modifier et enregistrer ladite base, un sous-procédé de détermination de la configuration permettant à un utilisateur d'une station de travail de construire une configuration désirée, un sous-procédé d'actualisation de la configuration désirée sur la station à configurer, un sous-procédé d'analyse et de contrôle de la configuration sur la station configurée et enfin un sous-procédé d'exploration de la configuration d'une station configurée à partir d'une station de travail.
6. Procédé de configuration de logiciels préconfigurés selon la revendication 5, caractérisé en ce que le sous-procédé d'édition de la base de connaissances, exécuté par le concepteur de la station de tra-

vail, utilise comme données d'entrée les données
 de la base de connaissances enregistrées dans la
 station de travail et introduites au travers d'une in-
 terface dite de connaissances ainsi que les don-
 nées de l'utilisateur destinées à modifier le contenu
 de la base de connaissances, les données de l'uti-
 lisateur étant entrées au travers de l'interface uti-
 lisateur graphique du concepteur qui est l'interface
 d'entrée pour le flux de données ainsi que l'interface
 de contrôle, les données de la base de connais-
 sances étant alors utilisées comme données de sortie,
 tandis que l'ensemble des fonctionnalités du con-
 cepteur et les données de l'utilisateur relatives à la
 configuration désirée sont utilisées pour, lors d'une
 première opération de création, exécuter le traite-
 ment propre à l'initialisation de la base de connais-
 sances, les données de sortie n'existant pas, la ba-
 se de connaissances étant vide, lors d'une seconde
 opération d'édition, prendre les données de la base
 de connaissances comme données d'entrée et
 fournir en sortie les données de la base de connais-
 sances éditées au travers de l'interface utilisateur
 graphique du concepteur, lors d'une troisième opé-
 ration de modification, prendre les données de la
 base de connaissances comme données d'entrée
 ainsi que les données de configuration désirée en-
 trées par l'utilisateur pour offrir en sortie la base de
 connaissances modifiée et enfin, lors d'une quatriè-
 me opération d'enregistrement, sauvegarder en en-
 registrant les données de la base de connais-
 sances.

7. Procédé de configuration de logiciels préconfigurés
 selon les revendications 5 et 6, caractérisé en ce
 que le sous-procédé de détermination de la confi-
 guration permet à un utilisateur de l'éditeur d'une
 station de travail de construire une configuration dé-
 sirée qui correspond à un ensemble hétérogène de
 plates-formes interconnectées au moyen de liaisons,
 cette configuration étant entrée au moyen de
 feuilles d'informations permettant d'identifier les
 composants associés à ces plates-formes, l'interfa-
 ce de contrôle étant l'interface utilisateur graphique
 de l'éditeur, tandis que l'ensemble des fonctionna-
 lités de l'éditeur, quelques fonctionnalités du con-
 cepteur pour l'accès à la base de connaissances
 ainsi que les données de l'utilisateur relatives à la
 configuration désirée sont utilisées pour, lors d'une
 première opération de création, exécuter le traite-
 ment relatif à l'initialisation de toute configuration,
 les données de sortie n'existant pas, la configura-
 tion étant vide, lors d'une seconde opération d'édi-
 tion, prendre le nom de la configuration entré par
 l'intermédiaire de l'interface utilisateur graphique de
 l'éditeur ainsi que les données de la base de con-
 naissances par l'intermédiaire de l'interface de con-
 naissances comme données d'entrée de même que
 l'ensemble des données correspondant à des con-

figurations déjà existantes desquelles il est extrait
 la configuration désirée par l'utilisateur, alors que
 les données fournies en sortie par l'intermédiaire de
 l'interface utilisateur graphique de l'éditeur corres-
 pondent à la configuration désirée par l'utilisateur,
 lors d'une troisième opération de modification de la
 topologie et des informations de configuration,
 prendre les données de configuration entrées par
 l'utilisateur par l'intermédiaire de l'interface uti-
 lisateur graphique de l'éditeur comme données d'en-
 trée, pour fournir une configuration modifiée éditée
 au travers de l'interface utilisateur graphique de
 l'éditeur, l'opération de modification de topologie
 fournissant la topologie et l'opération de modifica-
 tion des informations de configuration fournissant
 la topologie configurée et enfin, lors d'une quatriè-
 me opération de réalisation de la configuration,
 prendre la topologie configurée comme données
 d'entrée et éditer la configuration désirée par l'inter-
 médiaire de l'interface utilisateur graphique de l'édi-
 teur, ladite configuration étant enregistrée dans un
 ensemble de fichiers de configuration au travers de
 l'interface d'écriture des fichiers de configuration.

8. Procédé de configuration de logiciels préconfigurés
 selon les revendications 5 à 7, caractérisé en ce
 que le sous-procédé d'actualisation de la confi-
 guration désirée, est mis en oeuvre par l'interpréteur
 sur la station à configurer en faisant la différence
 entre une première configuration, une reconfigura-
 tion ou une déconfiguration, l'interpréteur étant éga-
 lement l'interface de contrôle alors que les données
 d'entrée et l'interface de ce sous-procédé sont
 constituées, en premier lieu, de fichiers de configu-
 ration décrivant d'une part, la configuration que le
 sous-procédé installe et d'autre part les éléments
 de la base de connaissances utilisés par l'interpré-
 teur, l'accès aux fichiers de configuration étant per-
 mis au travers d'une interface spécifique à l'inter-
 préteur, en second lieu, de la configuration effective
 du produit, conservée et accessible dans la station
 dans des fichiers de configuration spécifiques, l'in-
 terface permettant d'accéder à ces données d'en-
 trée étant constituée d'un ensemble de commandes
 spécifiques, en troisième lieu, du nom de la station
 à configurer entré par l'utilisateur, tandis que les
 données de sortie et l'interface de ce sous-procédé
 sont constituées d'une part, de la nouvelle configu-
 ration effective du produit qui, comme la précéden-
 te, est conservée et accessible dans la station dans
 des fichiers de configuration spécifiques, et d'autre
 part, d'enregistrements de la configuration effective,
 les données d'entrée et de sortie étant utilisées
 de telle manière que, lors d'une première opération
 de vérification de la configuration réellement instal-
 lée, il vérifie la totale cohérence entre la configura-
 tion effective de la station et les informations con-
 tenues dans les fichiers de configuration, lors d'une

seconde opération d'analyse de la configuration effective relativement aux contenus des fichiers de configuration, il compare la configuration effective de la station aux contenus des fichiers de configuration en générant les actions appropriées, et enfin lors d'une troisième opération d'activation de la configuration désirée, il lance la configuration à partir des données d'entrée constituées par les informations résultant de l'opération d'analyse précédente en installant les fichiers de configuration à la bonne place sur la station configurée et en exécutant les commandes de configuration récupérées lors de l'opération d'analyse précédente.

9. Procédé de configuration de logiciels préconfigurés selon les revendications 5 à 8, caractérisé en ce que le sous-procédé d'analyse et de contrôle de la configuration mis en oeuvre par l'interpréteur de la station configurée permet de récupérer la configuration de cette station et de fournir les informations de configuration dans un format aisément accessible au sous-procédé d'exploration de la configuration d'une station configurée à partir d'une station de travail, l'interpréteur étant également l'interface de contrôle alors que les données d'entrée et l'interface de ce sous-procédé sont constituées, d'une part, de la partie de la base de connaissances décrivant les stations, l'accès à cette base étant permis par l'intermédiaire d'une interface spécifique à l'interpréteur qui est la même que l'interface utilisée par le sous-procédé d'actualisation de la configuration désirée, et, d'autre part, de la configuration effective du produit, cette configuration étant conservée et accessible dans la station dans des fichiers de configuration spécifiques, l'interface permettant d'accéder à ces données d'entrée étant constituée d'un ensemble de commandes spécifiques, tandis que les données de sortie et l'interface de ce sous-procédé sont constituées des fichiers de configuration décrivant la configuration de la station analysée, les données d'entrée et de sortie étant utilisées de telle manière que, lors d'une première opération d'analyse de la station, il détermine la liste des composants installés sur la station analysée et récupère le nom et la version de chaque composant, les données d'entrée étant donc les données de la base de connaissances contenues dans les fichiers de configuration et les données de sortie étant la liste des composants, lors d'une seconde opération de lecture des données de configuration, il récupère les dites données de configuration propres aux produits installés sur la station, les données d'entrée étant donc les données de la base de connaissances et la liste des composants situés sur la station analysée fournie lors de l'opération d'analyse de la station et les données de sortie étant alors les données de configuration des produits présentées dans un format interne, et enfin lors d'une troisième

opération de création des fichiers de configuration, il prend comme données d'entrée les données de sortie générées lors de l'opération de lecture des données de configuration et, en premier lieu, pour chaque composant, construit l'ensemble des fichiers de configuration contenant les données de configuration dudit composant dans un format déterminé, en second lieu, pour l'ensemble des composants, construit l'arbre de tous les fichiers de configuration dans le format déterminé, alors qu'en troisième lieu, il enregistre cet arbre à un emplacement spécifié par l'utilisateur.

10. Procédé de configuration de logiciels préconfigurés selon les revendications 5 à 9, caractérisé en ce que le sous-procédé d'exploration de la configuration d'une station configurée mis en oeuvre par l'éditeur d'une station de travail permet d'éditer, sur cette station de travail, une configuration d'une quelconque station configurée et analysée en insérant la configuration de la station analysée dans une configuration éditée sur la station de travail, l'interface de contrôle étant l'interface utilisateur graphique de l'éditeur, tandis que l'ensemble des fonctionnalités de l'éditeur, quelques fonctionnalités du concepteur pour l'accès à la base de connaissances ainsi que les données de l'utilisateur relatives à la configuration désirée sont utilisées, alors que les données d'entrée sont, en premier lieu, les données de la base de connaissances enregistrée sur la station de travail, l'interface permettant d'accéder à cette base étant l'interface de connaissances, en second lieu, les fichiers de configuration générés par le sous-procédé d'analyse et de contrôle de la configuration sur la station configurée, lus au travers de l'interface de lecture des fichiers de configuration, et en troisième lieu, les structures de données spécifiant la configuration éditée dans laquelle la configuration de la station analysée est insérée, tandis que les données de sortie fournies par le sous-procédé d'exploration spécifient la configuration éditée dans laquelle la configuration de la station analysée a été ajoutée, les données d'entrée et de sortie étant utilisées de telle manière que, lors d'une première opération de lecture des données de configuration, il vérifie que la configuration est adaptée à la base de connaissances puis, remplit les structures de données internes avec les contenus des fichiers de configuration, les données d'entrée étant alors les données de la base de connaissances et les fichiers de configuration accédés au travers de l'interface de lecture des fichiers de configuration, lors d'une seconde opération de connexion dans le réseau représenté, il examine la configuration éditée ainsi que les données de configuration de la station analysée et détermine les liaisons permettant de connecter la station analysée aux autres stations présentes dans la configura-

ration éditée, les données d'entrée étant alors les structures de données représentant la configuration éditée et les structures de données représentant les données de configuration de la station analysée, cette opération de connexion modifiant les structures de données représentant la configuration éditée, lorsque les liaisons sont effectuées, et enfin lors d'une troisième opération d'affichage, il dessine la nouvelle topologie après insertion de la station analysée en incluant en premier lieu, la représentation graphique de la station analysée, en second lieu, la représentation graphique des liaisons permettant de connecter la station analysée au réseau, en troisième lieu, la représentation des données de configuration de la station analysée dans les feuilles d'informations, les données d'entrée étant alors les structures de données représentant la configuration éditée, modifiées par l'opération de connexion précédente.

20

25

30

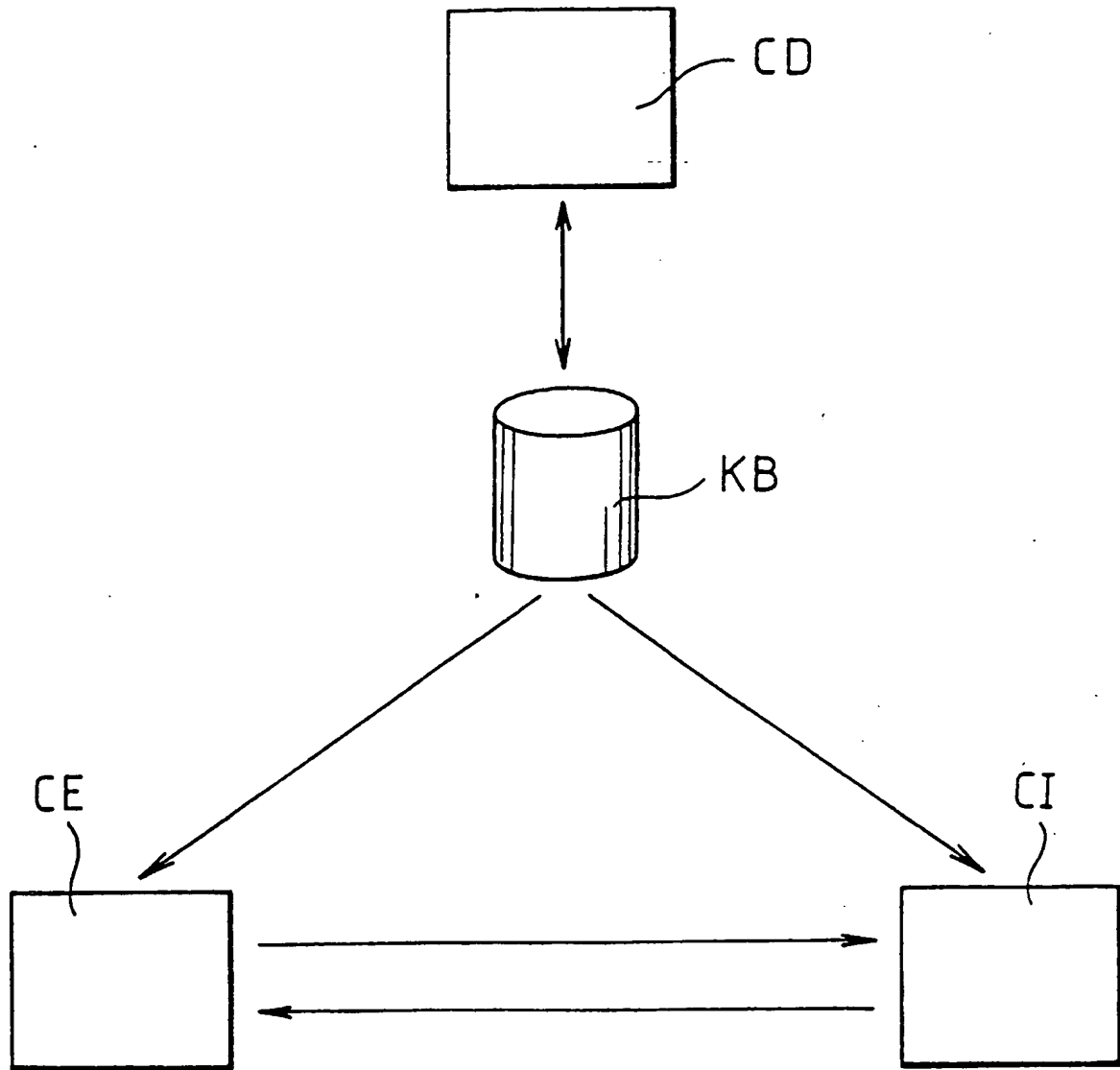
35

40

45

50

55



Office européen
des brevets

RAPPORT DE RECHERCHE EUROPEENNE

Numero de la demande
EP 97 40 0441

DOCUMENTS CONSIDERES COMME PERTINENTS			
Catégorie	Citation du document avec indication, en cas de besoin, des parties pertinentes	Revendication concernée	CLASSEMENT DE LA DEMANDE (Int.CL6)
X Y A	EP 0 398 644 A (IBM) 22 Novembre 1990 * abrégé * * colonne 1, ligne 40 - colonne 4, ligne 50 * * colonne 6, ligne 6 - colonne 12, ligne 20; figures 1,48 * ---	1 2,3 4-10	G06F9/445
Y	IBM TECHNICAL DISCLOSURE BULLETIN, vol. 34, no. 1, 1 Juin 1991, pages 430-434, XP000210264 "GRAPHIC APPC CONFIGURATION INTERFACE" * le document en entier * ---	2,3	
X A	EP 0 509 945 A (IBM) 21 Octobre 1992 * le document en entier * ---	1 2-10	
A	US 5 414 812 A (FILIP MICHAEL J ET AL) 9 Mai 1995 * colonne 5, ligne 40 - colonne 9, ligne 43 * ---	1-10	DOMAINES TECHNIQUES RECHERCHES (Int.CL6)
A	CHIP ZEITSCHRIFT FUER MIKROCOMPUTER-TECHNIK, no. 11, 1 Novembre 1991, pages 112-114, 116, XP000266204 VIETEN M: "AUF ZUM FROEHLICHEN FENSTERLN" * le document en entier * ---	3	G06F H04L
A	WO 95 14969 A (WANG LABORATORIES) 1 Juin 1995 * le document en entier * -----	2,3	
Le présent rapport a été établi pour toutes les revendications			
Lieu de la recherche LA HAYE		Date d'achèvement de la recherche 2 Juin 1997	Examineur Michel, T
CATEGORIE DES DOCUMENTS CITES		I : théorie ou principe à la base de l'invention E : document de brevet antérieur, mais publié à la date de dépôt ou après cette date D : cité dans la demande L : cité pour d'autres raisons & : membre de la même famille, document correspondant	
X : particulièrement pertinent à lui seul Y : particulièrement pertinent en combinaison avec un autre document de la même catégorie A : arrière-plan technologique O : divulgation non-écrite P : document intercalaire			

EPO FORM 1501 (01.02.1994)